REMARKS

Claims 1-33 are pending in the application. Claims 1, 2, 4-7, 9-14, 16-19, 21 and 23-30, 32-33 were rejected. Claims 3, 8, 15, 20, 22 and 31 are objected to.

The specification was objected to based on an informality. Applicants have corrected the informality.

Claim 27 was objected to based on an informality. Applicants have corrected the informality.

Figure 3 was objected to as requiring a "Prior Art" designation. Applicants have submitted herewith a corrected Figure 3.

Claims 1, 2, 4-6, 13, 14, 16-18, 27, 29-30 and 33 stand rejected under 35 U.S.C. § 103(a) as being unpatentable over Franaszek et al. (US 5,729,228) in view of Bigham (US 5,544,161).

In rejecting independent claims 1,13, and 27, the Office Action maintained that Franaszek teaches "the bitstream separated into blocks (b1 221, b2 222, b3 223, b4 224, called components)...." Office Action p. 3.

Applicants respectfully request reconsideration of the rejection in view of the following:

Applicants maintain that Claim 1, as amended, recites "separating a bitstream into a plurality of ***components of a pixel***; encoding the ***components***...."

Claim 13, as amended, recites "a plurality of encode units operable to receive components of a pixel separated from a bitstream and to encode the components using a compression algorithm...."

Applicants submit that the claimed separating "components of a pixel" for encoding in parallel is not taught or suggested by the prior art. The specification describes component in this context as "[p]ixels generally consist of one or more ***components***" See e.g., P. 17, ll. 7-8. Hence a component of a pixel is a subcomponent of a pixel (e.g., R, G, B, etc.).

By contrast, the references relied upon by the Examiner teach nothing of separately encoding components of pixels so that a pixel can have its components encoded in parallel, but rather merely teaches that "[b]lock B is logically divided into four equal size components, referred to as sub-blocks b1 (221), b2 (222), b3 (223), b4 (224)." Moreover, the fact that pixels "inherently include components" as indicated on page 4 of the office action does not teach or suggest separating pixels into components so that pixels components can be encoded in parallel.

For at least the foregoing reasons, Applicants respectfully request reconsideration of the 103(a) rejection with respect to independent claims 1 and 13. Applicants also respectfully request reconsideration of the rejection of dependent claims 2, 4-6, 14, and 16-18 by virtue of their respective incorporation of the limitations of the independent claims from which they depend.

Regarding claim 27, the Action notes that "Franaszek does not disclose expressly, scan lines, as recited by claim 27" and goes on to indicate that "scan lines inherently include pixels." Office Action p. 4. However, the Action does not provide a reference for the contention separating a bitstream into scanlines for parallel encoding is taught or suggested by the prior art. Accordingly, Applicants submit that claim 27 and its dependent claims 28-31 and 33 patentably define over the art of reference.

Claim 28 stands rejected under 35 U.S.C. § 103(a) as being unpatentable over Franaszek et al. in view of Bigham and further in view of Kuzma (USP 5,389,965).

Applicants submit that claim 28 defines over the references at least by virtue of its dependency from claim 27.

Claims 7, 9-12, 19, 21, and 23-26 were rejected under 35 USC 103(a) as being unpatentable over Franaszek et al. in view of Bigham and Schwartz et al.

Inasmuch as claims 7, 9-12, 19, 21, 23-26 and 32 depend from independent claims 1, 13, and 27, Applicants submit that they also define over the prior art at least for the reasons set forth above.

Applicants recognize that the Examiner has indicated that claims 3, 8, 15, 20, 22 and 31 present allowable subject matter and thank the Examiner for the thoughtful and considered examination. Applicants have rewritten dependent claims 3 and 15 in independent form as suggested by the Examiner. Applicants wish to pursue the remaining claims in their current dependent form. If the Examiner desires to discuss the remaining claims, Applicants' representative may be reached at **206.332.1386** and offers to discuss the pending claims at the Examiner's convenience to expedite the prosecution of this matter.

CONCLUSION

A Notice of Allowance for claims 1-33 is respectfully solicited.

Attached hereto is a marked-up version of the changes made to the specification and claims by the current amendment. The attached page is captioned **"Version with markings to show changes made."**

Respectfully submitted,

Date: August 9, 2002

Michael J. Swope
Registration No. 38,041

WOODCOCK WASHBURN LLP
One Liberty Place - 46th Floor
Philadelphia, PA 19103
Telephone: (215) 568-3100
Facsimile: (215) 568-3439

VERSION WITH MARKINGS TO SHOW CHANGES MADE

In the specification:

The paragraph beginning at line 5 of page 17, has been amended as follows:

With respect to component interleaving, fixed-length packetization can be [can be] done on a VxH rectangular region of pixels. Pixels generally consist of one or more components. The tag bits can be used to distribute the packets to different decoders, and the tag bits represent different scan lines. Within each scan line, the data can be encoded by interleaving blocks of components. This interleaving scheme can be the same one used for conventional LCF. Having the packetization scheme use the same interleaving pattern helps to simplify the logic that will be used to encode/decode both LCF and packetized LCF. For example, when encoding pixels which have four fully sampled components (RGBA, 4:4:.4:4:), there would be block of R, then a block of G, then a block of B, and finally a block of A before moving on to the next set of pixels. It should be noted that when encoding each block, the resulting encoded block could form a fraction of a packet, one packet, or multiple packets. The interleaving schemes can be as set forth in the following table.

In the claims:

Claims 1, 3, 13, 15 and 27 have been amended as follows:

1. A method for parallel compression and decompression of a bitstream, comprising:

separating a bitstream into a plurality of components of a pixel;

encoding the components using a compression algorithm;

constructing packets from the encoded components, where at least one packet is associated with each encoded component and the at least one packet comprises header information and encoded data;

combining the packets into a packetized encoded bitstream;

separating packets from the packetized encoded bitstream using the header information;

decoding packets in a parallel using a decompression algorithm to recover the encoded data;

constructing the plurality of components from the recovered encoded data; and

combining the plurality of components to recover the bitstream.

3. [The] A method [of claim 2, wherein] for parallel compression and decompression of a bitstream, comprising:

separating [the] a bitstream of a digitized graphic or video frame into a plurality of components [comprises] by separating the graphics or video frame into separate lines;

encoding the components using a compression algorithm;

constructing packets from the encoded components, where at least one packet is associated with each encoded component and the at least one packet comprises header information and encoded data;

combining the packets into a packetized encoded bitstream;

separating packets from the packetized encoded bitstream using the header information;

decoding packets in a parallel using a decompression algorithm to recover the encoded data;

constructing the plurality of components from the recovered encoded data; and

combining the plurality of components to recover the bitstream.

13. A system for parallel compression and decompression of a bitstream, comprising:

an encoder system comprising:

a plurality of encode units operable to receive components of a pixel separated from a bitstream and to encode the components using a compression algorithm;

the encode units further operable to construct packets from the encoded components, where at least one packet is associated with each encoded component and the at least one packet comprises header information and encoded data; and a multiplexer coupled to the encode units, the muliplexer operable to combine the packets into a packetized encoded bitstream; and

a decoder system comprising:

a feeder operable to separate packets from the packetized encoded bitstream;

a plurality of decode queues, the feeder further operable to distribute the packets to the decode queues;

a plurality of decode units each associated with one of the decode queues, the decode units operable to decode packets using a decompression algorithm to recover the encoded data and to reconstruct the components; and

a demultiplexer coupled to the plurality of decode units the demultiplexer operable to combine the plurality of components to recover the bitstream.

15. [The] A system [of Claim 14, wherein] for parallel compression and decompression of a bitstream, comprising:

an encoder system comprising:

plurality of encode units operable to receive a [the] plurality of components [comprise] comprising separate lines [of the] separated from a bitstream from a digitized graphics or video frame and to encode the components using a compression algorithm;

the encode units further operable to construct packets from the encoded components, where at least one packet is associated with each encoded component and the at least one packet comprises header information and encoded data; and

a multiplexer coupled to the encode units, the muliplexer operable to combine the packets into a packetized encoded bitstream; and

a decoder system comprising:

a feeder operable to separate packets from the packetized encoded bitstream;

a plurality of decode queues, the feeder further operable to distribute the packets to the decode queues;

a plurality of decode units each associated with one of the decode queues, the decode units operable to decode packets using a decompression algorithm to recover the encoded data and to reconstruct the components; and

a demultiplexer coupled to the plurality of decode units the demultiplexer operable to combine the plurality of components to recover the bitstream.

27. A method for parallel compression of graphic data, comprising:

separating a bitstream into a plurality of scan lines;

encoding each scan line into a plurality of blocks using a lossless compression algorithm; and

constructing at least one packet containing at least [on] one encoded block.